Constructions for Finite-State Codes

F. Pollara
Communications Systems Research Section

R. J. McEliece
California Institute of Technology and
Communications Systems Research Section

K. Abdel-Ghaffar
California Institute of Technology

In this article a class of codes called finite-state (FS) codes is defined and investigated. These codes, which generalize both block and convolutional codes, are defined by their encoders, which are finite-state machines with parallel inputs and outputs. A family of upper bounds on the free distance of a given FS code is derived, from known upper bounds on the minimum distance of block codes. A general construction for FS codes is then given, based on the idea of partitioning a given linear block code into cosets of one of its subcodes, and it is shown that in many cases the FS codes constructed in this way have a d_{free} *which is as large as possible. These codes are found without the need for lengthy computer searches, and have potential applications to future deep-space coding systems. The issue of catastrophic error propagation (CEP) for FS codes is also discussed, and it is found that, in order to avoid CEP, one must solve a very interesting problem in graph theory, the problem of finding a noncatastrophic edge-labeling of the state diagram.*

I. Introduction

Error-correcting codes are an essential part of all modern reliable and power-efficient deep-space communication systems. In this area of engineering, practice is currently leading theory, and as communication systems evolve, the new codes required must be found by elaborate computer searches. Such searches, although they often result in the discovery of powerful new codes (see e.g. [4]), are not wholly satisfactory for two reasons. First, computer searches are at present costly and time-consuming, and as communications systems evolve and the codes required become more and more complex, these searches may prove to be entirely impractical. Second, once a good new code is found by a search, there is rarely any guarantee that the best possible candidate has been identified. In this article we begin an attempt to remedy this problem by establishing a new theoretical framework for the simultaneous study of the two major classes of error-correcting codes, block and convolutional codes. It is our belief that this framework will allow researchers to construct provably optimal codes for use in future high-performance deep-space communication systems. The cornerstone of our theory is the notion of a finite state encoder, which we will now describe.

An (n, k, m) FS (finite state) encoder is a q^m-state finite state machine with k parallel inputs and n parallel outputs taken from a q-letter alphabet (Fig. 1). The encoder begins

from a fixed initial state. At each clock pulse, k symbols (the information symbols) are input to the encoder, and in response the encoder changes state and outputs n symbols (the code symbols). Thus if $(\mathbf{u}_1, \mathbf{u}_2, \ldots)$ is a sequence of k-symbol information blocks, then the encoder's output will be a sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots)$ of n-symbol code blocks, which we call a code sequence. The set of all such code sequences is called the code generated by the FS encoder. A code generated by a (n, k, m) FS encoder will be called an (n, k, m) finite state code. We note that if there is only one state in the encoder, the resulting $(n, k, 0)$ FS code is in fact an ordinary block code. Similarly, a linear convolutional code is just a FS code in which the finite-state machine is a bank of k parallel shift-registers, and each output symbol is a linear combination of the k input symbols and the symbols stored in the shift registers. Thus FS codes include both block and convolutional codes as special cases.

The free distance (d_{free}) of an FS code is defined to be the minimum Hamming distance between all pairs of distinct (infinite) code sequences. If the encoder isn't catastrophic, this is also the minimum Hamming distance between pairs of distinct finite code sequences, i.e., code sequences corresponding to distinct input sequences which lead the encoder from the initial state to the same final state. (If the encoder is catastrophic, it is possible that the smallest Hamming distance between two infinite code sequences could occur for a pair of paths through the encoder's state diagram which begin at the initial state but never again remerge. We will say more about catastrophic and noncatastrophic encoders in Section IV.)

In this article, our concern will be to find bounds on d_{free} in terms of the parameters n, k, and m, and to produce a family of FS codes meeting these bounds in certain cases. In Section II we derive our bounds; in Section III we describe a general construction for (n, k, m) FS codes, using ideas similar to those of Ungerboeck [3]; in Section IV we discuss the issue of catastrophic error propagation, and, using techniques from graph theory, describe an optimal noncatastrophic edge-labeling of the complete 2^m state diagram; and finally in Section V we combine the results of Sections III and IV to construct a class of Reed-Solomon-like FS codes which meet the bounds of Section II whenever $n \leqslant q$ and $m \leqslant \min(k-1, n-k-1)$.

II. Bounds on d_{free}

In this section we will derive a family of upper bounds on d_{free} in terms of the parameters n, k, and m. The basic idea is to find subcodes of a given FS code which are block codes, and use the fact that any upper bound on the minimum distance of the block code is also an upper bound on the free distance of the parent FS code.

Here is some needed notation: let $\Delta(n, k)$ denote the largest possible minimum distance for a block code over a q-letter alphabet with length n, and q^k code words. (We note for future reference the trivial fact that $\Delta(n, k)$ is meaningless for $k \leqslant 0$.) The following theorem gives a bound on the free distance of a FS code in terms of Δ.

Theorem 1. For any FS code with parameters n, k, and m, the free distance is bounded as follows:

$$d_{\text{free}} \leqslant \min_{L:Lk>m} \Delta(Ln, Lk-m)$$

Proof: We consider all possible input sequences consisting of L k-symbol input blocks $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_L)$. There are q^{Lk} such input sequences. For each of these sequences, the encoder starts in the initial state, and terminates in one of q^m states. It follows from the pigeon-hole principle that there must be at least q^{Lk-m} of these length-L input sequences which have the same final state. The code sequences corresponding to these input sequences can be thought of as a block code with length Ln, with at least q^{Lk-m} code words. The minimum distance of this block code is at most $\Delta(Ln, Lk-m)$, by definition. On the other hand, by the definition given in Section I, the minimum distance of this block code is an upper bound on the d_{free} of the original convolutional code. Since this is true for all L, we apparently have

$$d_{\text{free}} \leqslant \min_{L \geqslant 1} \Delta(Ln, Lk-m)$$

However, as we noted above, $\Delta(n, k)$ is meaningless, if $k \leqslant 0$, and so the minimization can only be taken over those values of L for which $Lk - m > 0$. ■

Corollary 1. The free distance of an (n, k, m) FS code over a q-letter alphabet satisfies

$$d_{\text{free}} \leqslant \min_{L:Lk>m} (Ln - Lk + m + 1)$$

$$= (n-k)\left\lfloor \frac{m}{k} + 1 \right\rfloor + m + 1$$

$$= n - k + 1 + m \qquad \text{if } k > m$$

Proof: This follows from Theorem 1 and the Singleton bound [2, Theorem 1.11], which says that

$$\Delta(n, k) \leqslant n - k + 1 \ ■$$

Corollary 2. The free distance of an FS code also satisfies

$$d_{\text{free}} \leq \min_{L:Lk>m} \left(Ln \frac{q-1}{q} \frac{q^{kL-m}}{q^{kL-m}-1} \right)$$

Proof: This follows from Theorem 1 and the Plotkin bound [1, Theorem 13.49], which says that

$$\Delta(n,k) \leq n \cdot \frac{q-1}{q} \cdot \frac{q^k}{q^k-1} \quad \blacksquare$$

III. Code Constructions

In the last section we derived bounds on d_{free} which apply to arbitrary FS codes. In this section, we will describe a very general construction for a class of FS codes; later in the article, we will find that many of the codes constructed by this technique meet the bounds of Section II.

Our basic idea is to start with an explicit state-transition diagram, and to build a code around this diagram. For definiteness, we will consider only complete q^m-state transition diagrams, in which every pair of states is connected by a directed edge (illustrated in Fig. 2 for $q = 2$, $m = 2$), but most of our ideas can be generalized to other diagrams.

The FS code is to have parameters n and k. This means that at every clock cycle k symbols go into the encoder and n symbols come out. Of the k input symbols, it is plausible to suppose that m are used to determine the next state of the encoder (recall that there are q^m states altogether) and $k - m$ are used to determine which n symbols are to be output. Thus it is natural to think of the possible n-symbol output blocks associated with a fixed state transition as the words in a $(n, k-m)$ block code. Our basic idea is to assign a $(n, k-m)$ block code to each possible state transition.

Here then is our general construction for an (n, k, m) FS code. We begin with an (n, k_1) block code C_1, with minimum distance d_1. We assume that C_1 can be decomposed into the disjoint union of a number of (n, k_2) subcodes, each with minimum distance d_2. (The easiest way, but not the only way, to arrange this decomposition is for C_1 to be a linear code, with an (n, k_2), $d_{\min} = d_2$ linear subcode. Then C_1 naturally decomposes into cosets of C_2. Each one of these cosets is then an (n, k_2) subcode with $d_{\min} = d_2$.) We assign one of the (n, k_2) subcodes to each of the q^{2m} state-transitions in the state transition diagram. We require that all q^m transitions originating at a given state, or terminating at a given state, be assigned a different subcode. This forces us to use at least q^m different subcodes; but in order to avoid catastrophic error propagation, we will need at least $2q^m$ subcodes, as we will see in Section IV. This requires that the dimensions of the big code C_1 satisfy

$$k_1 \geq k_2 + m + 1$$

The encoder now works as follows. Starting in the initial state, at every clock pulse it accepts $k = m + k_2$ input symbols. The first m of these symbols are used to determine the next state, and the remaining $k - m = k_2$ symbols are used to determine which of the q^{k_2} code words from the subcode corresponding to the state transition is to be output.

In the next theorem, we estimate the d_{free} of the code constructed in this way.

Theorem 2. The free distance of the $(n, m + k_2, m)$ FS code constructed as described above from (n, k_2), $d_{\min} = d_2$ subcodes of a (n, k_1), $d_{\min} = d_1$ block code satisfies

$$\min(d_2, 2d_1) \leq d_{\text{free}} \leq d_2$$

Proof: We need to estimate the Hamming distance between pairs of code sequences corresponding to paths in the state diagram which begin and end in the same state. Let us say that these paths both begin in state s_1 and end in state s_K. There are two cases to consider: (1) when the second states in the two paths are the same, and (2) when they are different (see Fig. 3).

In case (1) we look at only the first n-symbol block of each path. These two blocks are distinct code words in the same (n, k_2) subcode, and so they must differ in at least d_2 positions. Thus if case (1) holds, the Hamming distance between the two code sequences is at least d_2. Furthermore, since d_2 is the minimum distance of each of the subcodes, we know that the Hamming distance between some pair of code sequences is exactly d_2. Thus $d_{\text{free}} \leq d_2$.

In case (2) the paths must differ in at least two edges: the edges leaving s_1 and the edges next entering a common state (there must be such a common state since the paths both terminate at s_K; see Fig. 3). The n-symbol blocks corresponding to these pairs of edges are distinct code words in the (n, k_1) parent code, since we have assumed that different subcodes are assigned to all state transitions beginning, or ending in the same state, and so each pair must differ in at least d_1 positions. This means that the two code sequences must differ in at least $2d_1$ positions. Thus if case (2) holds, the Hamming distance between the two code sequences is at least $2d_1$.

Combining cases (1) and (2), we obtain the statement of the theorem. $\blacksquare$

Example 1: Let $q = 2$ and consider the 4-state diagram of Fig. 2. We choose as the parent code C_1 a $(16, 5)$ first-order Reed-Muller code with $d_1 = 8$. This code contains a $(16, 1)$ linear subcode C_2 (the repetition code) with $d_2 = 16$. There are 16 cosets of C_2 in C_1, and so it is possible to assign a different coset to each of the 16 state transitions in the state diagram. The result is a $(16, 3, 2)$ code with (according to Theorem 2) $d_{\text{free}} = 16$. On the other hand, by taking $L = 1$ in Corollary 2, we find that the free distance of a $(16, 3, 2)$ code with $q = 2$ is at most 16. Therefore the code constructed this way has the largest possible d_{free} for its given n, k, and m. (This example will be generalized in Example 4 in the next section.)

Example 2: We again take $q = 2$ and use the complete 4-state diagram of Fig. 2, but this time we take as the big code C_1 the $(16, 8)$ $d_{\min} = 6$ nonlinear Nordstrom-Robinson code. It is known ([2], Chapter 15) that this code is the union of 8 cosets of the $(16, 5)$ $d_{\min} = 8$ first-order Reed-Muller code. In the next section we will see that the edge-labeling given in Fig. 2 is noncatastrophic. Thus if we use the edge-labeling described in Fig. 2 to assign these 8 cosets to the 16 state transitions, Theorem 1 tells use that we get a $(16, 7, 2)$ FS code with $d_{\text{free}} = 8$. On the other hand, the bound in Corollary 2 (take $L = 1$) shows that any $(16, 7, 2)$ FS code must have $d_{\text{free}} \leqslant 8$, and so this code is optimum.

The codes constructed by the techniques of this section are often, as we have seen, quite good if d_{free} is used as the figure of merit. However, they are not yet guaranteed to be noncatastrophic. In the next section, we will address the problem of how to assign subcodes to state transitions to ensure noncatastrophicness.

IV. Noncatastrophic Edge Labelings

In the last section we showed how to construct an (n, k, m) FS code by assigning cosets of a subcode of an (n, k) block code to the edges of a state diagram. However, if the coset-to-edge assignments are not done carefully, the resulting encoder could be catastrophic. In this section, we will see how to make the coset assignment to avoid catastrophicness. We will see that catastrophicness can be avoided only if the number of cosets available is at least $2q^m$, and we will see one way to make a noncatastrophic coset-edge assignment if $2q^m$ cosets are available, and q is a power of 2.

We begin by saying what we mean by a noncatastrophic edge-labeling of a state diagram. If s and t are two states, we denote by $L(s, t)$ the label on the directed edge from s to t. (In our application, the "labels" are cosets.) Let

$$(s_1, s_2, \ldots, s_K) \quad \text{and} \quad (t_1, t_2, \ldots, t_K)$$

be two sequences of K states. If the two corresponding sequences of labels

$$L(s_1, s_2), L(s_2, s_3), \ldots, L(s_{K-1}, s_K)$$

and

$$L(t_1, t_2), L(t_2, t_3), \ldots, L(t_{K-1}, t_K)$$

are identical, we call two such state sequences label-indistinguishable.

Definition. An edge labeling of a state diagram is said to be noncatastrophic if and only if there is an integer K_0 such that any two label-indistinguishable state sequences of length $K \geqslant K_0$ are identical. (Informally, this says that if a state sequence is "long enough," it can be recovered uniquely from its label-sequence.)

A noncatastrophic edge-labeling guarantees that a bad burst of channel noise will never cause the decoder to make an infinite number of decoder errors, i.e., the decoder will not cause catastrophic error propagation. Notice that there exist catastrophic edge-labelings which do not cause catastrophic error propagation, but which also do not meet the condition for Theorem 2.

We can see how a catastrophic edge-labeling may cause catastrophic error propagation as follows. Let $(s_1, s_2, \ldots)$ and $(t_1, t_2, \ldots)$ be two arbitrarily long label-indistinguishable state sequences. If the encoder follows a state sequence that finishes with the sequence $(s_1, s_2, \ldots)$, it is possible for the channel noise to be such that the decoder will correctly determine the state sequence up to s_1, but then choose t_1 instead of s_1. If this happens, the decoder will almost surely never recover, since its metric calculations based on hypothesizing the incorrect state sequence $(t_1, t_2, \ldots)$ will be identical to those based on the true state sequence $(s_1, s_2, \ldots)$.

Notice that if all the edge labels are distinct, any state sequence can be uniquely recovered from its label sequence, and so the labeling must be noncatastrophic. (If the state diagram is the complete N-state diagram, this requires N^2 labels.) On the other hand, if all edges in the state diagram have the same label, all pairs of state sequences are label-indistinguishable, and so the labeling must be catastrophic. The basic problem is to find the minimum number of different labels that are needed for a noncatastrophic labeling. For an arbitrary state diagram, we do not know what this number is. However, if we assume that the state diagram is D-regular, i.e., there are exactly D edges coming in to and going out of each

state, the following theorem places a nontrivial lower bound on the required number of distinct labels.

Theorem 3. For a D-regular state diagram with at least two states, at least $2D$ distinct labels are required for a noncatastrophic edge labeling.

Proof: As a first step, note that if we have a labeling L such that $L(s, t) = L(s', t)$, where s and s' are distinct states, the labeling must be catastrophic, since then

$$(s, s_2, s_3, s_4, \ldots, s_K) \quad \text{and} \quad (s', s_2, s_3, s_4, \ldots, s_K)$$

are label-indistinguishable but not identical state sequences, for any value of K and for $s_2 = t$. Similarly, if $L(s, t) = L(s, t')$, where t and t' are distinct states, the labeling must also be catastrophic, since

$$(s_1, s_2, \ldots, s_{K-1}, t) \quad \text{and} \quad (s_1, s_2, \ldots, s_{K-1}, t')$$

are label-indistinguishable but not identical state sequences, for any value of K and for $s_{K-1} = s$. Thus we have for any noncatastrophic labeling that

$$L(s, t) \neq L(s', t) \qquad \text{if } s \neq s'$$

and

$$L(s, t) \neq L(s, t') \qquad \text{if } t \neq t'$$

In other words, if we are given x and $L(x, y)$, we can recover y; and if we are given y and $L(x, y)$, we can recover x. If the labeling has this property we will say that it is nonsingular. Thus all noncatastrophic labelings are nonsingular, but the converse may not hold.

Next we assume that we are given a noncatastrophic labeling of a D-regular state diagram, but that the labeling uses fewer than $2D$ labels. We will show by induction that this assumption leads to a contradiction, by constructing a pair of arbitrarily long nonidentical but label-indistinguishable state sequences. Since there are $D \times$ (no. of states) edges in the state diagram, but less than $2D$ labels, there must be two distinct edges with the same label. Denote these two edges by (s_1, s_2) and (t_1, t_2). This is a pair of label-indistinguishable but nonidentical state sequences of length 2. Assume that we have already constructed a pair of nonidentical but label-indistinguishable state sequences of length K, say

$$(s_1, s_2, \ldots, s_K) \quad \text{and} \quad (t_1, t_2, \ldots, t_K).$$

(We have just seen that we can do this for $K = 2$.) Since the labeling is nonsingular, we know that $s_K \neq t_K$. Now consider the $2D$ labels of the form $L(s_K, s)$ and $L(t_K, s)$. Since there are fewer than $2D$ labels available, two of these labels must be identical. These identical labels can't be of the form $L(s_K, s)$ and $L(s_K, s')$, or $L(t_K, t)$ and $L(t_K, t')$, since the labeling is nonsingular. Hence we must have $L(s_K, s) = L(t_K, t)$ for some s and t. Thus if we set $s_{K+1} = s$ and $t_{K+1} = t$, then

$$(s_1, s_2, \ldots, s_{K+1}) \quad \text{and} \quad (t_1, t_2, \ldots, t_{K+1})$$

are nonidentical but label-indistinguishable state sequences of length $K + 1$. This completes the proof that $2D$ labels are necessary in any noncatastrophic edge-labeling. ■

Theorem 3 says that $2D$ labels are necessary for a noncatastrophic labeling of a D-regular state diagram. In the next theorem, we will see that $2D$ labels are sufficient if D is a power of two, and if the state diagram is complete, i.e., every state is connected by a directed edge to every other state.

Theorem 4. The complete 2^m-state diagram can be labeled noncatastrophically with 2^{m+1} labels in such a way that every sequence of m edge labels uniquely identifies the state sequence.

Proof: Let us number the 2^m states with the integers in the set $\{0, 1, \ldots, 2^m - 1\}$. We will use the integers in the set $\{0, 1, \ldots, 2^{m+1} - 1\}$ as edge labels. Indeed, if x and y are two states, we label the directed edge from x to y with the integer $L(x, y) = y - 2x \bmod 2^{m+1}$. We claim that this labeling is noncatastrophic. As a first step in this direction, we note that this labeling is nonsingular. To see this, note that if we are given x and

$$L = y - 2x \bmod 2^{m+1}$$

then

$$y = L + 2x \bmod 2^{m+1}$$

and if we are given y and L, then

$$x = (y - L)/2$$

or

$$(y - L)/2 + 2^m$$

depending on which of these values is in the range $0 \leqslant x \leqslant 2^m - 1$.

Now to see why the labeling is noncatastrophic, let

$$x_0, x_1, \ldots, x_m \quad \text{and} \quad y_0, y_1, \ldots, y_m$$

be a pair of label-indistinguishable state sequences of length $m+1$, and let L_i be the common label on the edges $x_{i-1} \to x_i$ and $y_{i-1} \to y_i$. Then (all arithmetic is interpreted mod 2^{m+1}) we have

$$L_i = x_i - 2x_{i-1} = y_i - 2y_{i-1} \qquad \text{for } i = 1, 2, \ldots, m$$

From this we conclude that

$$x_1 = L_1 + 2x_0$$

$$x_2 = L_2 + 2L_1 + 4x_0$$

$$\vdots$$

$$x_m = L_m + \cdots + 2^{m-1} L_1 + 2^m x_0$$

Since $2^m x_0$ must be either 0 or 2^m (mod 2^{m+1}), and exactly one of

$$L_m + \cdots + 2^{m-1} L_1$$

and

$$L_m + \cdots + 2^{m-1} L_1 + 2^m$$

is in the range $\{0, 1, \ldots, 2^m\}$, it follows that x_m can be uniquely calculated from $L_1, L_2, \ldots, L_m$. Since y_m can be computed in exactly the same way, it follows that $x_m = y_m$. Since the labeling is nonsingular, it follows that

$$x_{m-1} = y_{m-1}, \ldots, x_1 = y_1$$

and so the two state sequences are identical. This proves that the given labeling is noncatastrophic, and indeed that any state sequence can be identified after at most m labels. ■

Example 3: The edge labels prescribed by the construction of Theorem 4 in the case $2^m = 4$ are given in Fig. 2; and in the case $2^m = 8$ are given in the following 8×8 matrix:

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	14	15	0	1	2	3	4	5
2	12	13	14	15	0	1	2	3
3	10	11	12	13	14	15	0	1
4	8	9	10	11	12	13	14	15
5	6	7	8	9	10	11	12	13
6	4	5	6	7	8	9	10	11
7	2	3	4	5	6	7	8	9

where the (x, y) entry of the matrix gives the label on the $x \to y$ state transition. Notice, for example, that the label sequence $(7, 4)$ is ambiguous (the state sequences $(0, 7, 5)$ and $(7, 5, 1)$ both yield the label sequence $(7, 4)$), but the label sequence $(7, 4, 11)$ uniquely specifies the state sequence $(0, 7, 5, 5)$.

We can combine Theorems 2 and 4 to give the following general construction for linear FS codes.

Theorem 5. Let q be a power of two, and suppose that C_1 is a (n, k_1), $d_{\min} = d_1$ block code over $GF(q)$, and C_2 is a (n, k_2), $d_{\min} = d_2$ subcode. Then there exists a $(n, k_1 - 1, k_1 - k_2 - 1)$ FS code with $\min(d_2, 2d_1) \leqslant d_{\text{free}} \leqslant d_2$.

Proof: Using the construction of Section III, we begin with a complete $q^{k_1 - k_2 - 1}$ state diagram. Since there are $q^{k_1 - k_2}$ cosets of C_2 in C_1, by Theorem 4 (since q is a power of two) it is possible to use these cosets to label the edges of the state diagram noncatastrophically. By Theorem 2 the result is a $(n, k_1 - 1, k_1 - k_2 - 1)$ FS code whose free distance satisfies the bounds given in the statement of the theorem. ■

Example 4 (Cf. Example 1): Let $q = 2$; let C_1 be the $(2^m, m+1)$, $d_{\min} = 2^{m-1}$ first-order Reed-Muller code, and let C_2 be the $(2^m, 1)$ $d_{\min} = 2^m$ repetition code, which is a subcode of C_1. Using Theorem 5 we can construct a $(2^m, m, m-1)$, $d_{\text{free}} = 2^m$ FS code, which by Corollary 2 to Theorem 1 (take $L = 1$) is optimal.

V. Reed-Solomon FS Codes

In this final section we will use the techniques of Sections II, III, and IV to construct a class of FS codes which are Reed-Solomon-like, and which meet the bounds of Section II in many cases.

The ancestors of the FS codes to be constructed are Reed-Solomon codes. We remind the reader that such codes are (n, k) block codes with minimum distance $d = n - k + 1$ (thus achieving the Singleton bound) and that these codes exist for all n and k satisfying $1 \leqslant k \leqslant n \leqslant q$, where q is the alphabet size (see [2], Chapter 10). To simplify the construction, we will assume in what follows that the alphabet size q is a power of two.

Our goal is to use a Reed-Solomon code to construct an (n, k, m) FS code with $d_{\text{free}} = n - k + 1 + m$, which by Corollary 1 is the largest possible value. Following the prescription in Section III, we let C_1 be an $(n, k+1)$ Reed-Solomon code with $d_1 = n - k$. The subcode C_2 is taken to be an $(n, k - m)$ Reed-Solomon code, with $d_2 = n - k + 1 + m$ (this requires

$m < k$). Thus by Theorem 5, the resulting (n, k, m) code will have

$$\min(2(n-k), n-k+1+m) \leqslant d_{\text{free}} \leqslant n-k+1+m$$

If $2(n-k) \geqslant n-k+1+m$, i.e., $m \leqslant n-k-1$, this gives $d_{\text{free}} = n-k+1+m$, and by Corollary 1 the free distance cannot be larger than this. Therefore we have proved the following.

Theorem 6. For any parameters n, k, m, and q (q must be a power of 2) satisfying $k \leqslant n-1 \leqslant q-1$ and

$$m \leqslant \min(k-1, n-k-1)$$

there exists a noncatastrophic (n, k, m) FS code whose free distance meets the Singleton bound, viz.

$$d_{\text{free}} = n-k+1+m$$

Example 5: If we start with a (15, 11) RS code over $GF(16)$, a code with $d_{\min} = 5$, (thus $k = 10$ in the construction described above), we can construct $(15, 10, m)$ FS codes, for $0 \leqslant m \leqslant 10$. For $m = 0, 1, 2, 3$, and 4, the codes have $d_{\text{free}} = m + 6$, which agrees with the Singleton bound of Corollary 1, and so these codes are all optimal. For $5 \leqslant m \leqslant 10$, however, the codes constructed all have $d_{\text{free}} = 10$, independent of m, do not meet the Singleton bound, and presumably do not have the largest possible d_{free}'s for their values of n, k, and m.

VI. Conclusion

In this article we have introduced the notion of a finite state code in an attempt to unify the theory of block and convolutional codes and to establish a theoretical framework which will allow researchers to explicitly construct powerful new error-correcting codes for deep-space and other applications. Although the results in this article are highly promising, and some of the codes we have constructed are likely to be useful in some applications, much further work remains to be done. In particular, the specific constructions given in Section III only scratch the surface of the interesting problem of synthesizing good FS codes. The central problem here is to take a good block code and to partition it into a disjoint union of isomorphic good subcodes. Another problem worthy of further research is that of finding good noncatastrophic edge labelings of specific state diagrams. Finally, we have not addressed the important problem of decoding at all. And while it seems that many FS codes can be decoded practically using a combination of Viterbi's algorithm with a corresponding block decoding algorithm, this question certainly needs serious study if FS codes are to be used in practice.

References

[1] E. R. Berlekamp, *Algebraic Coding Theory* (Revised 1984 ed.). Laguna Hills, Calif.: Aegean Park Press, 1984.

[2] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.

[3] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Information Theory*, vol. IT-28, Jan. 1982, pp. 55–67.

[4] J. H. Yuen and Q. D. Vo, "In search of a 2-dB coding gain," *TDA Progress Report 42-83*, vol. July–Sept. 1985, pp. 26–33, Jet Propulsion Laboratory, Pasadena, Calif., November 15, 1985.

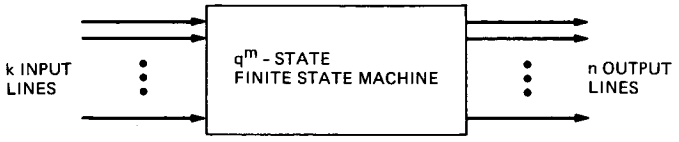

Fig. 1. A Finite-State encoder

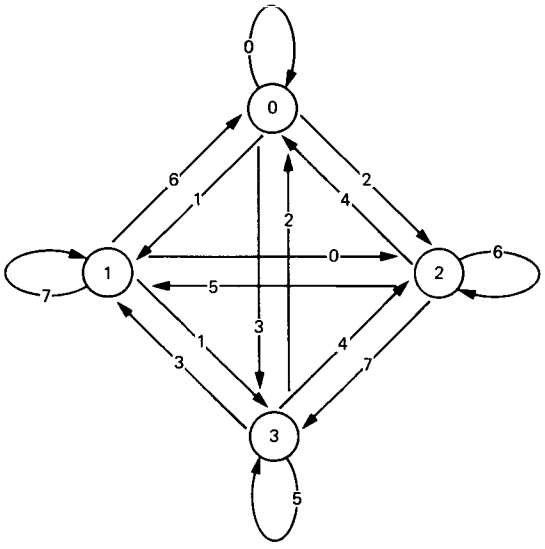

Fig. 2. A complete 4-state diagram. The edge labels are as described in Theorem 4 in Section IV.

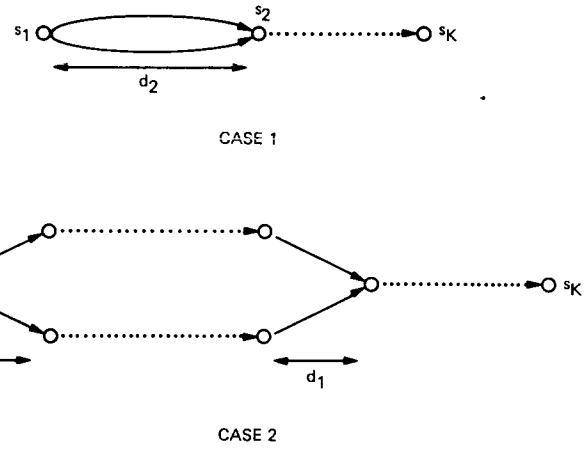

Fig. 3. The proof of Theorem 2